

Amendments to the Claims:

Please cancel Claim 22 and amend Claims 1, 19, 23 and 24 as indicated below. This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (currently amended) An apparatus comprising:

a micro-operation generator to receive a predicated instruction that indicates a plurality of destination registers;

the micro-operation generator further to generate a plurality of micro-operations that implement the instruction;

the micro-operation generator further to decompose the instruction into the plurality of micro-operations such that each of the micro-operations indicates a single destination register that is distinct from the destination registers indicated by the other micro-operations.

2. (original) The apparatus of claim 1, wherein

the plurality of micro-operations includes one or more standard micro-operations to provide the basic functionality indicated by the instruction.

3. (original) The apparatus of claim 1, wherein

the plurality of micro-operations includes one or more predication-support micro-operations.

4. (original) The apparatus of claim 3, wherein:

the one or more predication-support micro-operations includes a conditional move micro-operation for each destination register indicated by the instruction, where the conditional

move micro-operation is to select between a first and second proposed output value based on the value of a qualifying predicate associated with the instruction.

5. (original) The apparatus of claim 4, wherein:

the first proposed output value is the incoming value of the associated destination register; and

the second proposed output value is a computed value.

6. (original) The apparatus of claim 4, wherein:

the first proposed output value is the incoming value of the associated destination register; and

the second proposed output value is a constant value.

7. (original) The apparatus of claim 6, wherein the instruction is a compare instruction.

8. (original) The apparatus of claim 7, wherein the instruction is an “or” form compare instruction.

9. (original) The apparatus of claim 7, wherein the instruction is an “and” form compare instruction.

10. (original) The apparatus of claim 7, wherein the instruction is an “and”-complement form compare instruction.

11. (original) The apparatus of claim 7, wherein the instruction is an “or”-complement form compare instruction.

12. (original) The apparatus of claim 4, wherein:
the instruction is a parallel compare instruction.

13. (original) The apparatus of claim 12, wherein:

the one or more predication-support micro-operations include a first conditional move micro-operation for a first destination register indicated by the parallel compare instruction;

the one or more predication-support micro-operations also include a second conditional move micro-operation for a second destination register indicated by the parallel compare instruction;

the first proposed output value for the first conditional move micro-operation is the incoming value of the associated destination register;

the second proposed output value for the first conditional move micro-operation is a first constant;

the first proposed output value for the second conditional move micro-operation is the incoming value of the associated destination register;

the second proposed output value for second first conditional move micro-operation is a second constant.

14. (original) The apparatus of claim 13, wherein the instruction is an “and.orcm” form of a parallel compare instruction.

15. (original) The apparatus of claim 12, wherein the instruction is an “or.andcm” form of a parallel compare instruction.

16. (original) The apparatus of claim 3, wherein:

the one or more predication support micro-operations includes an append operation to append a qualifying predicate value to a proposed output value.

17. (original) The apparatus of claim 3, wherein:

the instruction is a load instruction; and

the one or more predication support micro-operations includes an append operation to append a qualifying predicate value to a load address.

18. (original) The apparatus of claim 3, wherein:

the instruction is a store instruction; and

the one or more predication support micro-operations includes an append operation to append a qualifying predicate value to a store address.

19. (currently) A system comprising:

a memory to store an instruction that indicates a plurality of destination registers; and

a processor coupled to the memory;

wherein the processor includes a microcode engine to generate, for the instruction, a series of micro-operations, wherein each of the micro-operations indicates a single distinct destination operand and a maximum of two source operands.

20. (original) The system of claim 19, wherein:

the memory is a DRAM.

21. (original) The system of claim 19, wherein:

the instruction is predicated; and

the micro-code engine is further to generate, for each of the destination registers indicated by the instruction, a micro-operation to conditionally move one of two proposed values into the destination register.

22. (Cancel)

23. (currently amended) The system of claim ~~[[22]]~~21, wherein:

a micro-operation to store an incoming value of the destination register is further to append a qualifying predicate value for the instruction with the stored incoming value.

24. (currently amended) A method comprising:

receiving an instruction that indicates a plurality of destination registers; and

generating a plurality of standard micro-operations for the instruction, wherein each of the standard micro-operations indicates a single destination register that is distinct from the destination registers indicated in of the other standard micro-operations.

25. (original) The method of claim 24, further comprising:

generating, if the instruction is predicated, a plurality of predication-support micro-operations for the instruction.

26. (original) The method of claim 25, wherein generating a plurality of predication-support micro-operations further comprises:

generating a conditional move micro-operation for each destination register indicated by the instruction.

27. (original) The method of claim 26, wherein generating a conditional move operation further comprises:

if the instruction indicates a known constant as a first potential output value, generating a micro-operation that receives a second potential output value as an input, wherein execution of the micro-operation is to cause a processor to move either the constant value or a second potential output value into a destination register.

28. (original) The method of claim 26, wherein generating a conditional move operation further comprises:

if the instruction does not indicate a known constant as a potential output value, generating a micro-operation that receives a first potential output value as an input and also

receives a second potential output value as an input, wherein execution of the micro-operation is to cause a processor to move either the first potential output value or the second potential output value into a destination register.

29. (original) The method of claim 25, wherein generating a plurality of predication-support micro-operations further comprises:

generating an append micro-operation for each destination register indicated by the instruction, wherein execution of the micro-operation is to cause a processor to append a qualifying predicate for the instruction with a potential output value, and is to further cause the appended value to be stored in a register.

30. (original) An apparatus comprising:

a micro-operation generator to receive a predicated instruction, the predicated instruction being one of a plurality of instructions, the plurality including arithmetic instructions and memory instructions;

the micro-operation generator further to generate a plurality of micro-operations that implement the instruction.

31. (original) The apparatus of claim 30, wherein the predicated instruction is an arithmetic instruction.

32. (original) The apparatus of claim 31, wherein the predicated instruction is an add instruction.

33. (original) The apparatus of claim 31, wherein the predicated instruction is a compare instruction.

34. (original) The apparatus of claim 30, wherein the predicated instruction is a memory instruction.
35. (original) The apparatus of claim 34, wherein the predicated instruction is a load instruction.
36. (original) The apparatus of claim 34, wherein the predicated instruction is a store instruction.
37. (original) The apparatus of claim 30, wherein
- the plurality of micro-operations includes one or more standard micro-operations to provide the basic functionality indicated by the instruction.
38. (original) The apparatus of claim 30, wherein
- the plurality of micro-operations includes one or more predication-support micro-operations.
39. (original) The apparatus of claim 38, wherein:
- the one or more predication-support micro-operations includes a conditional move micro-operation for each destination register indicated by the instruction, where the conditional move micro-operation is to select between a first and second proposed output value based on the value of a qualifying predicate associated with the instruction.
40. (original) The apparatus of claim 39, wherein:

the first proposed output value is the incoming value of the associated destination register; and

the second proposed output value is a computed value.

41. (original) The apparatus of claim 39, wherein:

the first proposed output value is the incoming value of the associated destination register; and

the second proposed output value is a constant value.

42. (original) The apparatus of claim 33, wherein the instruction is an “or” form compare instruction.

43. (original) The apparatus of claim 33, wherein the instruction is an “and” form compare instruction.

44. (original) The apparatus of claim 33, wherein the instruction is an “and”-complement form compare instruction.

45. (original) The apparatus of claim 33, wherein the instruction is an “or”-complement form compare instruction.

46. (original) The apparatus of claim 31, wherein:
the instruction is a parallel compare instruction.

47. (original) The apparatus of claim 38, wherein:

the one or more predication-support micro-operations include a first conditional move micro-operation for a first destination register indicated by a parallel compare instruction;

the one or more predication-support micro-operations also include a second conditional move micro-operation for a second destination register indicated by the parallel compare instruction;

the first proposed output value for the first conditional move micro-operation is the incoming value of the associated destination register;

the second proposed output value for the first conditional move micro-operation is a first constant;

the first proposed output value for the second conditional move micro-operation is the incoming value of the associated destination register; and

the second proposed output value for second first conditional move micro-operation is a second constant.

48. (original) The apparatus of claim 46, wherein the instruction is an “and.orcm” form of a parallel compare instruction.

49. (original) The apparatus of claim 46, wherein the instruction is an “or.andcm” form of a parallel compare instruction.

50. (original) The apparatus of claim 38, wherein:

the one or more predication support micro-operations includes an append operation to append a qualifying predicate value to a proposed output value.

51. (original) The apparatus of claim 38, wherein:

the instruction is a load instruction; and

the one or more predication support micro-operations includes an append operation to append a qualifying predicate value to a load address.

52. (original) The apparatus of claim 38, wherein:

the instruction is a store instruction; and

the one or more predication support micro-operations includes an append operation to append a qualifying predicate value to a store address.

53. (original) The apparatus of claim 30, wherein:

the micro-operation generator is further to decompose the instruction into the plurality of micro-operations such that each of the micro-operations indicates a single destination register.